Perspectives in Science (2016) 7, 126—132

ScienceDirect

journal homepage: www.elsevier.com/pisc

A novel memristive cellular neural network with time-variant templates☆

Xiaofang Hu[a], Guanrong Chen[a,*], Shukai Duan[b]

[a] Department of Electronic Engineering, City University of Hong Kong, Hong Kong
[b] College of Electronics and Information Engineering, Southwest University, Chongqing 400715, China

Received 26 October 2015; accepted 11 November 2015
Available online 11 December 2015

KEYWORDS
Cellular neural network (CNN);
Memristor;
Time-variant template;
Edge detection;
Noise reduction

Summary A cellular neural network (CNN) is a massively parallel analog array processor capable of solving various complex processing problems by using specific templates that characterize the synaptic connections. The hardware implementation and applications of CNN have attracted a great deal of attention. Recently, memristors with nanometer-scale and variable gradual conductance have been exploited to make compact and programmable electric synapses. This paper proposes and studies a novel memristive CNN (Mt-CNN) with time-variant templates realized by memristor crossbar synaptic circuits. The template parameters are estimated analytically. The Mt-CNN provides a promising solution to hardware realization of real-time template updating processes, which can be used to effectively deal with various complicated problems of cascaded processing. Its effectiveness and advantages are demonstrated by practical examples of edge detection on noisy images.

Introduction

A cellular neural network (CNN) (Chua and Yang, 1988a,b) possesses the neuromorphic property of local connectivity with a topographic array of simple processing cells, thus suitable for very-large-scale integration (VLSI) chip implementation (Adamatzky et al., 2013). Executing various tasks by using specific cloning templates at a high speed, more than a thousand times faster than regular digital processors (Zarándy, 1999), makes the CNN widely adopted in image and video processing, biological and robotic version, and a variety of other real-time applications (Chua and Roska, 2002). Furthermore, CNN with time-variant templates can execute cascaded processing operations in a one-layer array via real-time template updating. Therefore, it is quite suitable for image processing of multiple tasks (Harrer, 1993).

However, although some small operational CNN chips have been developed (Cruz and Chua, 1998; Dominguez-Castro et al., 1997), their VLSI implementation by using the conventional CMOS-based integration technology remains a big challenge. Even if CNN hardware circuit can be built,

☆ This article is part of a special issue entitled ''Proceedings of the 1st Czech-China Scientific Conference 2015''.

* Corresponding author.
E-mail addresses: xiaofanhu2@cityu.edu.hk (X. Hu), eegchen@cityu.edu.hk (G. Chen), duansk@swu.edu.cn (S. Duan).

http://dx.doi.org/10.1016/j.pisc.2015.11.021

its synaptic connections cannot be easily updated. That is another factor restricting the practical use of CNN chips today.

The memristor was predicted by Chua in 1971 (Chua, 1971) and physically developed by HP researchers in 2008 (Strukov et al., 2008). Possessing desirable characteristics such as small size, automatic memory and low power consumption, the memristor has been incorporated into a variety of technologies, including the nonvolatile memory (Hu et al., 2012; Zidan et al., 2013), ultra-high density Boolean logic and signal processing (Borghetti et al., 2010), as well as nonlinear circuits (Hu et al., 2014), to name just a few. In particular, memristors exhibit a remarkable similarity to biological synapses in variable conductance subject to external excitations and thus become one prospective candidate of electric synapses in neuromorphic computing systems (Alibart et al., 2012; Alibart et al., 2013). Notably, memristors have also been proposed to realize the templates of CNNs (Corinto et al., 2014; Duan et al., 2014; Kim et al., 2012; Wen et al., 2014; Yilmaz and Mazumder, 2013). Apart from circuit compactness and nonvolatility, the greatest advantage of memristive templates lies in its good programmability. Utilizing these special advantages of memristors, our objective in this paper is to design an efficient compact memristive CNN with time-variant templates.

Specifically, in this paper a compact memristive CNN (Mt-CNN) model with time-variant templates is proposed and studied. A memristor-based synaptic circuit is used to realize the time-variant template where the template parameters are estimated analytically. Furthermore, its application to edge detection of noisy images is discussed. Illustrative examples and comparative analysis show its effectiveness and advantages over its traditional counterparts.

Memristor synaptic circuit

Preliminaries

The acclaimed HP titanium dioxide memristor can be modelled by a combination of two variable resistors in series made respectively from two materials of pure TiO_2 with very low conductivity and oxygen-deficient TiO_{2-X} with much higher conductivity which are sandwiched in between two metal electrodes (e.g. Pt) (Hu et al., 2014; Kim et al., 2012; Strukov et al., 2008). Its operating mechanism is primarily described through the change of the conductance state caused by the movement of the doping front between these two material regions under externally applied stimuli.

The relationship between the input u and the output y of the memristor is described by

$$y = M\left(\frac{w}{D}\right)u, \quad y = G\left(\frac{w}{D}\right)u, \tag{1}$$

for the current-controlled case, namely $u=i$ and $y=v$, or for the voltage-controlled case with $u=v$ and $y=i$. Here, w and D are the width of the TiO_{2-X} region and the overall titanium dioxide thin-film, respectively. The memristance (memristor-resistance, M) and memconductance (memristor-conductance, G) are respectively defined by

$$M\left(\frac{w}{D}\right) = \frac{w}{D}R_L + \left(1-\frac{w}{D}\right)R_H, \tag{2}$$

$$G\left(\frac{w}{D}\right) = \frac{G_L G_H}{(w/D)G_H + (1-(w/D))\,G_L}, \tag{3}$$

where $R_L(G_L = {R_L}^{-1})$ and R_H $(G_H = {R_H}^{-1})$ denote the limiting resistance (conductance) when the doping front at the boundaries of the device is at $w=D$ or $w=0$, respectively. The width w is chosen as the internal state variable and its dynamics are described by

$$\dot{w} = i\frac{\mu_v R_L}{D} f(w, v) \tag{4}$$

where μ denotes the average mobility of the oxygen vacancies, the window function $f(w,u)$ is an estimation of the oxygen vacancy drift. In this paper, the window function associated with the so-called BCM (boundary condition memristor) model is adopted because of its simple, accurate and flexible expression (Corinto and Ascoli, 2012), as

$$f(w, v) = \begin{cases} \gamma, & \text{if } C_1 \text{ or } C_2 \text{ holds,} \\ 0, & \text{if } C_3 \text{ or } C_4 \text{ holds,} \\ \alpha, & \text{if } C_5 \text{ holds,} \end{cases} \tag{5}$$

where $\gamma \in \mathrm{R}_{0,+}$ and $\alpha \in \mathrm{R}_{0,+}(\gamma > \alpha)$ describe the degree of nonvolatility of the device, and

$$\begin{cases} C_1 = & \left\{w \in (0, D)\,\&\,((v > v_{to}) \text{ or } (v < -v_{t1}))\right\}, \\ C_2 = & \left\{(w = 0\,\&\,v > v_{tho}) \text{ or } (w = D\,\&\,v < -v_{th1})\right\}, \\ C_3 = & \left\{w = 0\,\&\,v \le v_{tho}\right\}, \\ C_4 = & \left\{w = D\,\&\,v \ge -v_{th1}\right\}, \\ C_5 = & \left\{w \in (0, D)\,\&\,(v \le v_{to})\,\&\,(v \ge -v_{t1})\right\}. \end{cases} \tag{6}$$

It is noted that $v_{tho} \in \mathrm{R}_{0,+}$ and $v_{th1} \in \mathrm{R}_{0,+}$ denote the boundary thresholds for releasing the memristor state from its limiting values near the device boundaries, and $v_{to} \in \mathrm{R}_{0,+}$ and $v_{t1} \in \mathrm{R}_{0,+}$ represent the programmability thresholds, namely the magnitudes that an input voltage needs to exceed in order to provide a (much) bigger velocity of the state change.

The memristor synaptic circuit

In artificial neural networks, the weighted summation of input data and synaptic weights is the most critical part and also a key issue hindering their VLSI implementation. In the Mt-CNN design, the synaptic connection is realized by a crossbar-based memristor synaptic circuit, as shown in Fig. 1, where the weighted summation can be executed in an efficient compact manner.

In this synaptic circuit, voltages $V_1, V_2, \ldots, V_k$ are either the memristor conductance programming excitations or the input data to be weighted by the memristive weights. Each pair of memristors represents a synaptic weight and the weighting operation is carried out in a simple way via Ohm's Law. Taking for example a pair of memristors with conductance G_i^+ and G_i^-, respectively, the synaptic weight that they represent is given by

$$\omega_i \propto G_i \equiv G_i^+ - G_i^-, \tag{7}$$

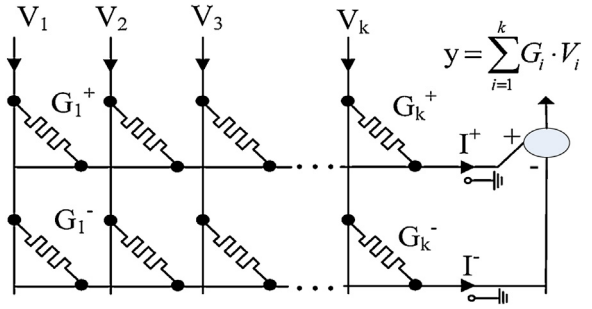

Figure 1 Memristor-based implementation of the synaptic weight and the weighted summation.

which can take either a positive or a negative value. Then, the weighted currents are summed up in an analogue fashion, as

$$I = I^{+} - I^{-} = \sum_{i=1}^{k} (G_i^{+} \cdot V_i - G_i^{-} \cdot V_i) \propto \sum_{i=1}^{k} \omega_i V_i. \tag{8}$$

Two of such memristor crossbar synaptic circuits can realize the programmable template parameters.

It is noted that there is an important issue in the design of memristor-based synaptic circuits; that is, how to ensure the desirable linear weighting when the neuromorphic system is at the working mode. Thanks to the activation threshold property of the memristor, the weighting can be performed approximately linearly by setting the working signals (input data) to be smaller than the activation threshold.

The Mt-CNN with time-variant templates

An $M \times N$ memristive cellular neural network (Mt-CNN) consists of MN cells and the interconnections among the cells in every neighbourhood. The neighbourhood set $N_r(i, j)$ of an inner cell (not on any boundary including its corners) is defined as

$$N_r(i,j) = \{c(i,j) | \max_{1 \le k \le M, 1 \le l \le N} (|k-i|, |j-1|) \le r|\} \tag{9}$$

where r is a positive integer denoting the neighbourhood radius, and pairs (i, j) and (k, l) are the positions of cells $c(i, j)$ and $c(k, l)$ on the grid of the Mt-CNN, respectively. A cell circuit of the Mt-CNN includes one linear resistor, one linear capacitor, one independent current source (threshold current), one independent voltage source (input), one piecewise-linear voltage-controlled current source representing the state-to-output transformation, and two linear voltage-controlled current sources satisfying $I_{xy}(i, j; k, l) = A_m(i, j; k, l)y_{kl}$ and $I_{xu}(i, j; k, l) = B_m(i, j; k, l)u_{kl}$ for all $c(k, l) \in N_r(i, j)$, similarly to the conventional CNN (Chua and Yang, 1988b). The dynamical equations of an Mt-CNN cell are described as follows:

$$\begin{cases} \dot{x}_{ij}(t) = -x_{ij}(t) + \sum\limits_{c(k,l) \in Nr(i,j)} A_m(i,j;k,l) y_{kl}(t) + \sum\limits_{c(k,l) \in Nr(i,j)} B_m(i,j;k,l) u_{kl} + z_{ij}, \\ y_{ij}(t) = \dfrac{1}{2}(|x_{ij}(t)+1| - |x_{ij}(t)-1|), \end{cases} \tag{10}$$

where $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$, $x(t)$, $y(t)$ and $u(t)$ denote the state, output and input, respectively, and z_{ij} is the threshold; matrices A_m and B_m represent the memristive

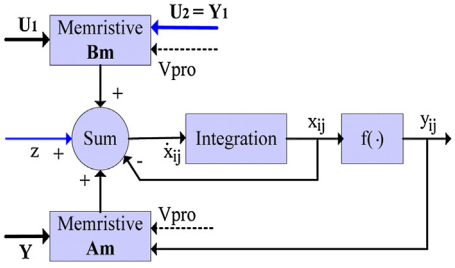

Figure 2 Conceptual structure of an Mt-CNN cell.

time-variant feedback and feed-forward cloning templates, respectively. They are realized by the memristor synaptic circuits (7) and (8).

Template design for edge detection

In this section, an application example on edge detection of noisy images is used to describe the operating mechanism of the Mt-CNN. Fig. 2 shows a conceptual structure of the Mt-CNN cell, which corresponds to (10). The processing procedure of the Mt-CNN is divided into two stages. Firstly, the memristors are programmed by programming voltage V_{pro} to realize the noise reduction template, where the initial input of the system is the noisy image (U1). After a period of dynamical iterations, the process enters into the next stage. Then, the intermittent output (Y1), namely the noise-reduced image, becomes the new input (U2). The memristors are updated to realize the edge detection template for executing the edge detection operation, where one important criterion for determining if a stage can be terminated could be the convergence ratio. Since both the input data and the initial state of the CNN belong to the range $[-1, 1]$, to avoid confusion the condition $|x(t)| > 1$ is chosen as the indication of convergence of a cell. Thus, the convergence ratio of the network can be defined by

$$c_x = \frac{n_x}{MN} \tag{11}$$

where n_x is the number of cells that have reached stable states and MN is the total cell number of the Mt-CNN. Set a threshold for the convergence ratio. Then, once its convergence ratio reaches this threshold, the network is considered having completed the current task, and so the iteration can stop.

The template design using a mathematical analysis method as presented in (Chua and Yang, 1988b; Li et al., 2008) was usually under the assumption that ''+1'' corresponds to ''black'' and ''−1'' corresponds to ''white'', which is opposite to the principle of conventional image displays. In this paper, a modified template design for edge detection Mt-CNN is proposed, in which ''+1'' and ''−1'' directly represent ''white'' and ''black'', respectively. The

edge detection template and the threshold can be described in the following forms:

$$A_m = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_m = \begin{bmatrix} -c & -c & -c \\ -c & b & -c \\ -c & -c & -c \end{bmatrix}, \quad Z = z \tag{12}$$

where a, b and c are positive real numbers and z is a real number. The template can be estimated based on some global tasks and some local rules modified from (Li et al., 2008), as follows:

Global tasks	Local rules
1) Given: Binary or grey-scale image P. 2) Input: U = P. 3) Initial state: $X(0) = 0$, 1 or -1. 4) Output: $Y(t) \to Y(\infty)$ = binary image where the ''black pixels (-1)'' correspond to pixels lying on sharp edge of P and the ''white pixels (+1)'' represent the background and non-edge pixels.	For each cell $c(i, j)$, given input u_{ij}, the final output $y_{ij}(\infty)$ will take either ''+1'' or ''-1'' under certain conditions, as follows: u_{ij} and $y_{ij}(\infty)$. 1) White $\to$ white, independent of the neighbours, 2) Black $\to$ white, if all nearest neighbours are black, 3) Black $\to$ black, if at least one nearest neighbour is white, 4) Grey $\to$ black, if the Laplacian operator gives $\nabla^2 U_{ij} < -z$, 5) Grey $\to$ white, if the Laplacian operator gives $\nabla^2 U_{ij} > -z$, 6) Grey $\to$ 0, if the Laplacian operator gives $\nabla^2 U_{ij} = z$.

In the above operations, the Laplacian operator is defined by

$$\nabla^2 U_{i,j} = \sum_{c(k,l) \in Nr(i,j)} B_m(i, j; k, l) u_{kl} \tag{13}$$

The CNN can detect edges of binary and grey-scale images in an efficient way, where the parameters of the feedback template are zeros, except possibly the center element. This CNN is also called an uncoupled CNN. In this case, the state equation of the uncoupled Mt-CNN cell can be rewritten as

$$\dot{x}_{ij}(t) = -x_{ij}(t) + a y_{ij}(t) + g(t) \tag{14}$$

where

$$g(t) = b u_{ij} - c \sum_{k,l \neq i,j} u_{k,l} + z \tag{15}$$

where $a = A_m(i, j; i, j)$ (self-feedback), and both of the state resistor (R_x) and the linear capacitor (C) are set to be one.

Lemma 1 ((Chua and Yang, 1988b)). If the circuit parameter satisfies $A(i, j; i, j) > {R_x}^{-1}$, where A is the feedback template, then the cell state will settle at a stable equilibrium point after the transient has decayed to zero. Moreover, the magnitudes of all stable equilibrium points are greater than one.

Since R_x is assumed to be one here, the above sufficient condition can be rewritten as $a > 1$. This lemma can be proved by using a graph-theoretic method based on the driving-point plots shown in Fig. 3(a). Specifically, the blue and pink-red lines denote the cases of $a < 1$ and $a > 1$, respectively. Herein, the solid curves are yielded by setting $g(t) = 0$ and the dashed and dotted curves represent the cases of $g(t) > 0$ and $g(t) < 0$, respectively. It can be seen that no matter what value $g(t)$ takes, when $a < 1$, the dynamic system has only one equilibrium; while when $a > 1$, it may have two stable equilibria falling into some regions beyond $[-1, 1]$. Moreover, according to the saturated characteristic of the output function in (10), the condition $a > 1$ can guarantee binary output.

The relationship between the system dynamics and $g(t)$ is further analyzed based on the driving-point curves plotted in Fig. 3(b), with $a > 1$. From top to bottom, the five curves respectively denote the situations when $g(t)$ takes five values from different ranges. Based on (14) and (15) and the output function, the final output of the Mt-CNN can be characterized as follows:

$$y_{ij}(\infty) = \begin{cases} 1, & g(t) \geq a - 1, \quad x_{ij}(0) \in (-\infty, \infty), \\ 1, & 1 - a < g(t) < a - 1, \quad x_{ij}(0) > -\dfrac{\omega_{ij}}{a - 1}, \\ -1, & 1 - a < g(t) < a - 1, \quad x_{ij}(0) < -\dfrac{\omega_{ij}}{a - 1}, \\ -1, & g(t) \leq 1 - a, \quad x_{ij}(0) \in (-\infty, \infty). \end{cases} \tag{16}$$

Since the initial state $x_{ij}(0) \in \{-1, 0, 1\}$, (16) can be rewritten in a more compact form, as

$$y_{ij}(\infty) = \begin{cases} 1, & g(t) \geq a - 1, \\ -1, & g(t) \leq 1 - a. \end{cases} \tag{17}$$

Now, based on (17) as well as the global tasks and local rules, the design criteria for the edge detection template of the Mt-CNN are obtained as follows.

Theorem 1. For $x_{ij}(0) \in \{-1, 0, 1\}$, or more generally $x_{ij}(0) \in (-\infty, \infty)$, the Mt-CNN satisfies the local rules if the following inequalities hold:

$$\begin{cases} a > 1, \\ b - 8c + z \geq \mathrm{a} - 1, \\ -b + 8c + z \geq a - 1, \\ -b + 6c + z \leq 1 - a. \end{cases} \tag{18}$$

Proof. By local rule (1): if $u_{ij} = 1$ and $y_{ij}(\infty) = 1$, then based on (17), one has

$$g(t) = b u_{ij} - c \sum_{k,l \neq i,j} u_{k,l} + z \geq b - 8c + z \geq a - 1 \tag{19}$$

By local rule (2): if $u_{ij} = -1$, all nearest neighbours are black, and $y_{ij}(\infty) = 1$, then

$$g(t) = -b + 8c + z\omega_{ij} \geq a - 1 \tag{20}$$

By local rule (3): if $u_{ij} = -1$, at least one nearest neighbour is white, and $y_{ij}(\infty) = -1$, then

$$g(t) = -b + 6c + z \leq 1 - a \tag{21}$$

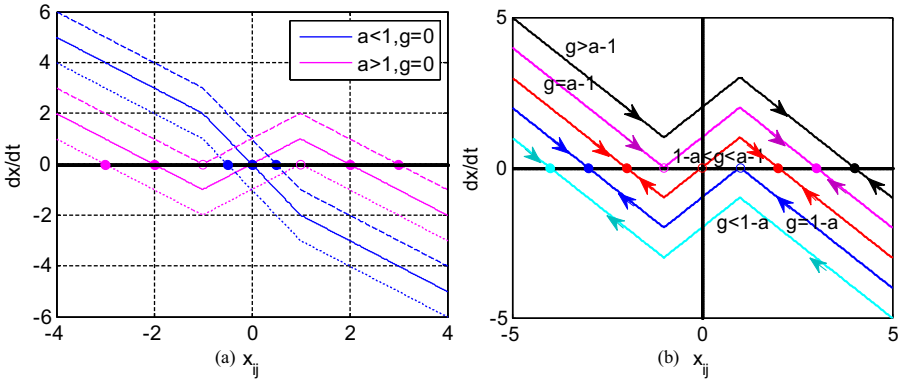

Figure 3 Driving-point plots of the Mt-CNN. (a) From top to bottom on the left of the origin: $a < 1$ and $g(t) > 0$, $a < 1$ and $g(t) = 0$, $a < 1$ and $g(t) < 0$, $a > 1$ and $g(t) > 0$, $a > 1$ and $g(t) = 0$, $a > 1$ and $g(t) < 0$, respectively; (b) $a > 1$ and $g(t) > a - 1$, $g(t) = a - 1$, $1 - a < g(t) < a - 1$, $g(t) = 1 - a$, $g(t) < 1 - a$.

By local rule (4): if $|u_{ij}| < 1$, $\nabla^2 U_{ij} \leq -z + 1 - a$, and $y_{ij}(\infty) = -1$, then

$$g(t) = \nabla^2 U_{ij} + z \leq 1 - a \tag{22}$$

By local rule (5): if $|u_{ij}| < 1$, $\nabla^2 U_{ij} \geq -z + a - 1$, and $y_{ij}(\infty) = 1$, then

$$g(t) = \nabla^2 U_{ij} + z \geq a - 1 \tag{23}$$

By local rule (6): if $|u_{ij}| < 1$, $\nabla^2 U_{ij} = -z$, and $y_{ij}(\infty) = 0$, then

$$g(t) = \nabla^2 U_{ij} + z = 0 \tag{24}$$

Combining (19)—(24) proves the theorem.

Experiments and results

Some numerical simulations carried out on Matlab are now presented, so as to illustrate the effectiveness of the

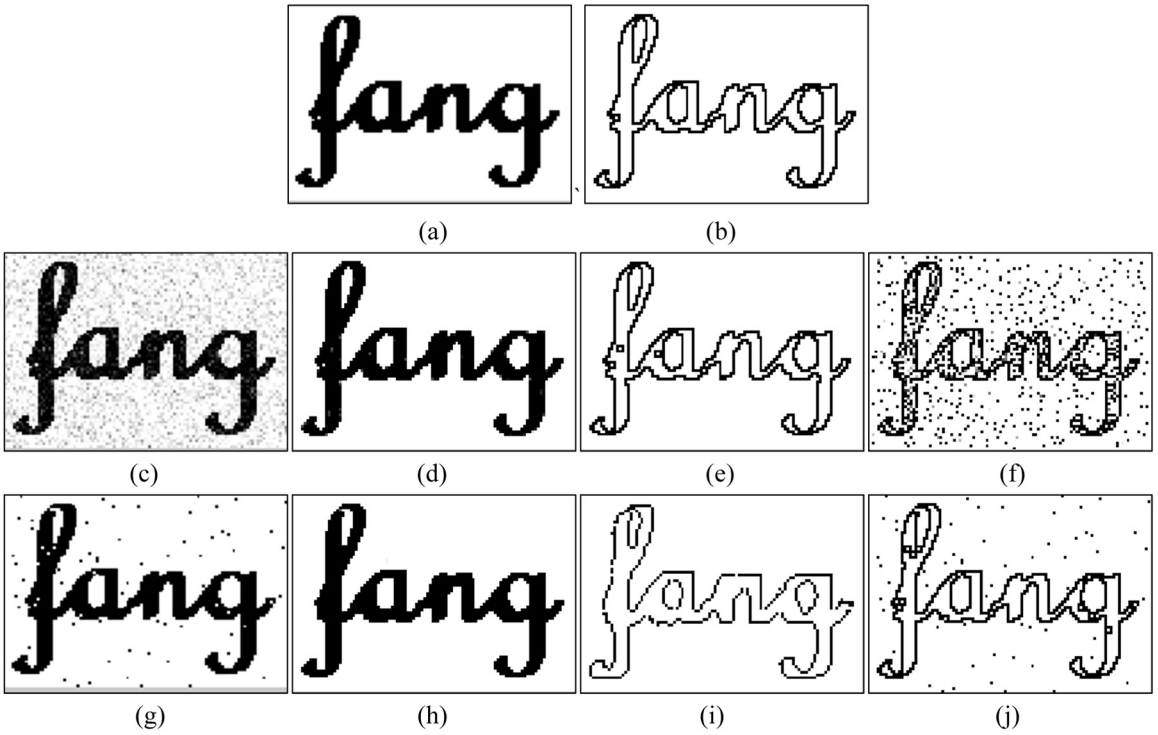

Figure 4 Edge detection of noisy images via the Mt-CNN. (a) Ideal image; (b) edge detection result of the ideal image; (c) noisy image with Gaussian noise ($\sigma = 0.02$); (d) noise reduction result of image (c); (e) edge detection result of noise-reduced image (d); (f) edge detection result of noisy image (c); (g) noisy image with ''salt and pepper'' noise (2%); (h) noise reduction result on image (g); (i) edge detection result on noise-reduced image (h); (j) edge detection result on noisy image (g).

proposed scheme. The memristor employed in the simulations is the aforementioned BCM model. During the weighting and the weighted summation operations, the input voltage is set to be lower than the activation threshold, e.g. pixel intensity ''+1'' and ''−1'' are represented by $V_+ = 0.6$ and $V_- = -0.6$, respectively.

Then, edge detection of two noisy images via the Mt-CNN is illustrated, and the results are presented in Fig. 4. Herein, the templates for noise reduction and those for edge detection are given, respectively, as follows:

$$A_{mG} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad B_m = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad B_m, Z = 0,$$

$$A_m = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_m = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad Z = 0.5$$

For comparison, Fig. 4(a) and (b) show the original ideal image and its edge detection result by the Mt-CNN. Fig. 4(c)−(e) are noisy image input with Gaussian noise ($\sigma = 0.02$), noise-reduced result (first stage) and final edge detection result, respectively. Here, the convergence threshold is chosen to be 95%. In comparison, if the detection process is directly executed on the noisy image, the resulting image would still carry some noise, sometimes even become worse, as illustrated by Fig. 4(f).

In addition to being effective on Gaussian-noise removal, the Mt-CNN also works well for other kinds of noise. Fig. 4(g) shows an image contaminated with ''salt and pepper'' noise (2%), for which the corresponding results are presented in Fig. 4(h)−(j), respectively.

It is noted that since the noise reduction is often imperfect, the quality of the results shown in Fig. 4(e) and (f) is somewhat lower than the ideal case, but still much better than the results obtained without noise reduction.

Conclusions

In this paper, we propose and study a memristive CNN with time-variant templates. Memristors are employed to construct the microscopic crossbar-based synaptic circuit for realizing the time-variant templates, which provides a promising approach to the hardware implementation of real-time template updating. In addition, an edge detection template is estimated with the coherent image representation based on the common display principle. Furthermore, the effectiveness of the proposed Mt-CNN is verified via an application example of edge detection for noisy images.

This work is expected to contribute to the further development of the CNN with time-variant templates in efficient VLSI implementation and practical applications, such as skeletonization, rectangular hull extraction, halftoning, and colour image processing.

Conflict of interest

The authors declare that there is no conflict of interest.

Acknowledgments

The work was supported by the Hong Kong Research Grants Council under the Grant CityU11201414.

References

Adamatzky, A., Chen, G., Chaos, C.N.N., 2013. Memristors and Beyond: A Festschrift for Leon Chua. World Scientific Publishing Co., Inc.

Alibart, F., Gao, L., Hoskins, B.D., Strukov, D.B., 2012. High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm. Nanotechnology 23 (7), 075201.

Alibart, F., Zamanidoost, E., Strukov, D.B., 2013. Pattern classification by memristive crossbar circuits using ex situ and in situ training. Nat. Commun. 4, 1−7.

Borghetti, J., Snider, G.S., Kuekes, P.J., Yang, J.J., Stewart, D.R., Williams, R.S., 2010. 'Memristive' switches enable 'stateful' logic operations via material implication. Nature 464 (7290), 873−876.

Chua, L.O., 1971. Memristor-the missing circuit element. IEEE Trans. Circuit Theory 18 (5), 507−519.

Chua, L.O., Roska, T., 2002. Cellular Neural Networks and Visual Computing: Foundations and Applications. Cambridge University Press.

Chua, L.O., Yang, L., 1988a. Cellular neural networks: applications. IEEE Trans. Circuits Syst. 35 (10), 1273−1290.

Chua, L.O., Yang, L., 1988b. Cellular neural networks: theory. IEEE Trans. Circuits Syst. CAS-35 (10), 1257−1272.

Corinto, F., Ascoli, A., 2012. A Boundary Condition-Based Approach to the Modeling of Memristor Nanostructures.

Corinto, F., Ascoli, A., Kim, Y.-S., Min, K.-S., 2014. Cellular Nonlinear Networks with Memristor Synapses Memristor Networks. Springer, pp. 267−291.

Cruz, J.M., Chua, L.O., 1998. A 16×16 cellular neural network universal chip: the first complete single-chip dynamic computer array with distributed memory and with gray-scale input−output. Analog Integr. Circuits Signal Process. 15 (3), 227−237.

Dominguez-Castro, R., Espejo, S., et al., 1997. A 0.8-μm CMOS two-dimensional programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage. IEEE J. Solid-State Circuits 32 (7), 1013−1026.

Duan, S., Hu, X., Dong, Z., Wang, L., Mazumder, P., 2014. Memristor-based cellular nonlinear/neural network: design, analysis, and applications. IEEE Trans. Neural Netw. Learn. Syst. (99), 1−12, http://dx.doi.org/10.1109/TNNLS.2014.2334701.

Harrer, H., 1993. Multiple layer discrete-time cellular neural networks using time-variant templates. IEEE Trans. Circuits Syst. II: Analog Digital Signal Process. 40 (3), 191−199.

Hu, X., Chen, G., Duan, S., Feng, G., 2014. A memristor-based chaotic system with boundary conditions. In: Chua, A.A.a.L.O. (Ed.), Memristor Networks. Springer, pp. 351−364.

Hu, X., Duan, S., Wang, L., Liao, X., 2012. Memristive crossbar array with applications in image processing. Sci. China Inf. Sci. 55 (2), 461−472.

Kim, H., Sah, M., Yang, C., Roska, T., Chua, L.O., 2012. Neural synaptic weighting with a pulse-based memristor circuit. IEEE Trans. Circuits Syst. I: Regular Papers 59 (1), 148−158.

Li, G., Min, L., Zang, H., 2008. Color edge detections based on cellular neural network. Int. J. Bifurc. Chaos 18 (04), 1231−1242.

Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S., 2008. The missing memristor found. Nature 453 (7191), 80−83.

Wen, S., Zeng, Z., Huang, T., Yu, X., 2014. Noise cancellation of memristive neural networks. Neural Netw. 60, 74—83.

Yilmaz, Y., Mazumder, P., 2013. Image processing by a programmable grid comprising quantum dots and memristors. IEEE Trans. Nanotechnol. 12 (6), 879—887.

Zarándy, Á., 1999. The art of CNN template design. Int. J. Circuit Theory Appl. 27 (1), 5—23.

Zidan, M.A., Fahmy, H.A.H., Hussain, M.M., Salama, K.N., 2013. Memristor-based memory: the sneak paths problem and solutions. Microelectron. J. 44 (2), 176—183.